Graph Attention in Attention Network for Image Denoising

Bo Jiang, Yao Lu*, Xiaosheng Chen, Xinhai Lu, and Guangming Lu*, *Senior Member, IEEE*

***Abstract*—Image denoising aims to remove the noise from noisy images. With the increasing complexity of the noise within the noisy images, current denoising methods can not satisfactorily address this issue. This paper proposes a Graph Attention in Attention Network (GAiA-Net) for image denoising. Firstly, we introduce a novel approach to graph construction for the GAiA-Net. In the process of such graph construction, the noisy images are divided into patches to formulate the nodes in a graph. The edges are initialized using k-nearest neighbors. Hence, through iterative transformation and learning, both the pixel-level and structure-level features can be captured by different information exchanges and aggregation within (pixel-level) and outside (structure-level) of the nodes, respectively. Secondly, we propose the Graph Attention in Attention (GAiA) in the GAiA-Net. The proposed GAiA produces the pixel-level attention within nodes to be further induced to the nodes with various distances to generate the final attention. Therefore, our GAiA-Net can capture the long dependencies on both the pixel-level and structure-level features, which can effectively reduce the complex noise in the denoising process. Comprehensive experiments demonstrate that the proposed GAiA-Net produces state-of-the-art performances on both synthetic noise image and real noise image datasets. Especially, when experimenting on complex noisy Nam datasets, our GAiA-Net achieves a PSNR of 40.40 dB and SSIM of 0.989. These results prove the satisfactory potential and effectiveness of our GAiA-Net.**

I. Introduction

IMAGE denoising aims to recover the high-quality images from their noisy image observations. Image denoising is an ill-posed problem, because there are multiple denoised image solutions for noisy image input. To address this inverse problem, some classical non-local methods [1], [2], [3], [4] capture the correlation between non-local self-similar blocks from the perspective of internal image-specific information [5] to improve image denoising performance. With the development of deep learning, some denoising methods based on **C**onvolutional **N**eural **N**etworks (CNNs) [6], [7], Transformer structures [8], [9], and **G**raph **C**onvolution **N**etworks (GCNs) [10], [11] have been proposed to capture various priors by learning the mappings between noisy and clean images from external training data. However, due to the increasing severe complexity of noise within the noisy images, current deep denoising methods still can not satisfactorily remove the noise and recover the clean images. These deep denoising methods will be explored in the following paragraphs.

Although conventional CNNs [6], [7] have large receptive fields, they fail to capture long-range (non-local self-similar) dependencies. GCN-based [10] and Transformer-based [12] image denoising methods can capture long-range dependencies. However, they still seriously suffer from many issues. We compare the GCN-based and Transformer-based methods in Fig. 1. The specific analyses are demonstrated as follows. As shown in Fig. 1(a), GCN-based methods [13], [11] treat each pixel in the image as a node, and the edges in the graph are computed by Edge-Condition Convolution (ECC). The above graphs are aggregated to capture long-range dependencies from the pixel-level perspective to reconstruct denoised images. However, from the feature maps produced by the GCNs-based methods, it is obvious that using simple ECC to predict edges between nodes may ignore structure-level features. Therefore, ***(a) GCN-based methods can retrieve long-range dependencies mainly on the pixel-level feature, which may lose insufficiently global structure-level feature, and thus leads to artifacts in the reconstructed denoised images.***

As shown in Fig. 1(b), the Transformer-based methods [14], [8], [9] treat the noisy image mechanically as a sequence of image patches, and aggregate the sequence of image patches through the self-attention mechanism from the perspective of structure-level feature to remove the noise in the noisy image. From the feature maps generated by the Transformer-based methods, using the self-attention mechanism to mix image patches can significantly extract structure-level features. However, due to the lack of pixel-level features, the lines or contours in the generated feature maps are quite weak, resulting in smeared textures in the reconstructed denoised images. Hence, ***(b) Transformer-based methods apply the self-attention mechanism to capture long-range dependencies of image patches mainly on the structure-level feature, while ignoring the refinement of pixel-level feature, which may cause the smeared textures in the recovered denoised images.***

This work was supported in part by the NSFC fund 62176077 and 62206073, in part by the Shenzhen Key Technical Project under Grant 2022N001, 2020N046, and 2022N063, in part by the Shenzhen Fundamental Research Fund under Grant JCYJ20210324132210025, in part by the Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies (2022B1212010005), in part by the Guangdong Shenzhen joint Youth Fund under Grant 2021A151511074, in part by the Natural Science Foundation of Guangdong Province under Grant 2023A1515010893, and in part by the Shenzhen Doctoral Initiation Technology Plan under Grant RCBS20221008093222010.

*Yao Lu and Guangming Lu are the corresponding authors.

Bo Jiang is with the Department of Computer Science, Harbin Institute of Technology at Shenzhen, Shenzhen 518057, China (e-mail: jiangbo_PhD@outlook.com).

Yao Lu and Guangming Lu are with the Department of Computer Science and Technology, Harbin Institute of Technology at Shenzhen, and Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies, Shenzhen 518057, China (e-mail: luyao2021@hit.edu.cn; luguangm@hit.edu.cn)

Xiaosheng Chen and Xinhai Lu are with the Shenzhen Second People's Hospital, Shenzhen 518057, China (e-mail: dr_chenxiaosheng@163.com; luxinhai@szsqsnjzjkzx.wecom.work).

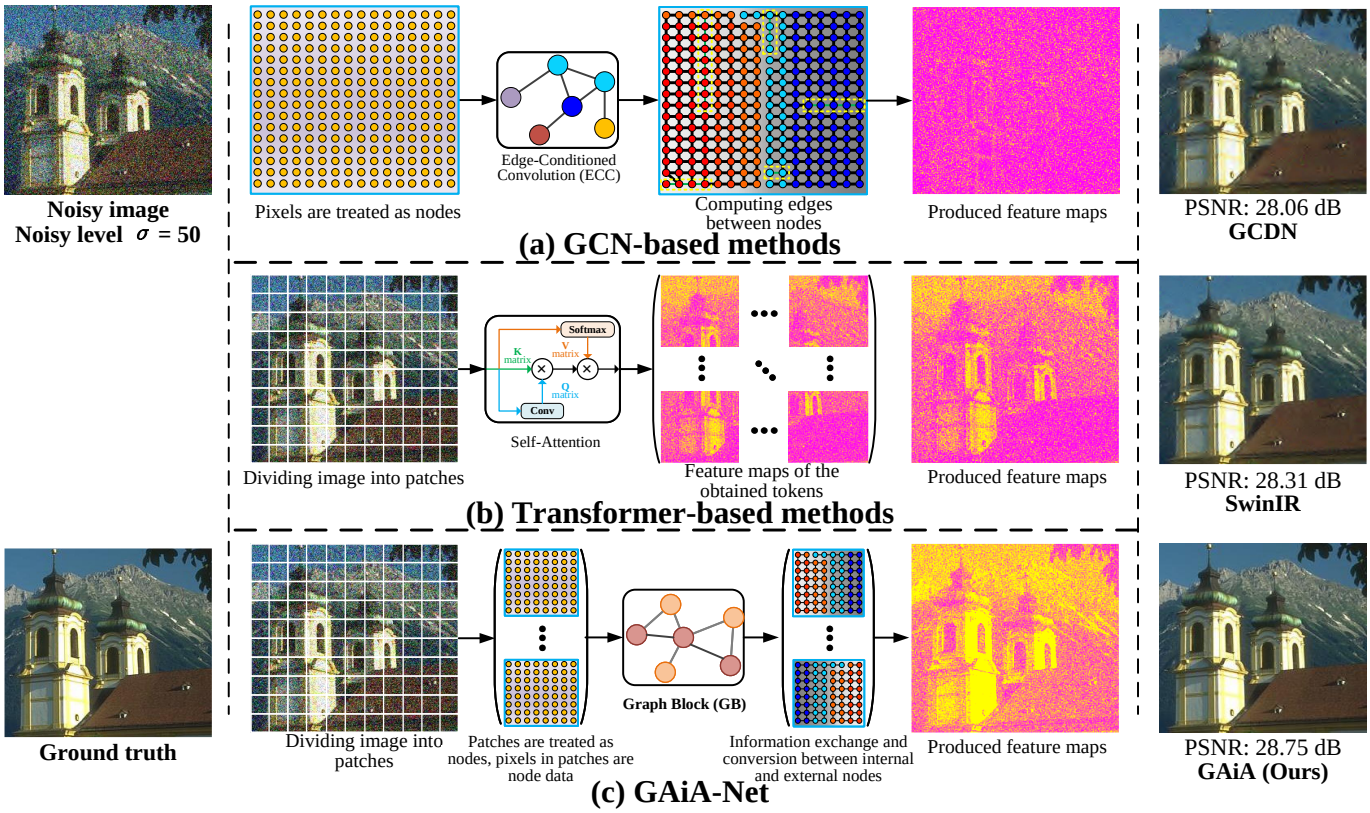

Fig. 1. Schematic of piplines based on different strategies. (a) is pipeline of the GCN-based methods, capturing long-range dependencies from the pixel-level feature perspective. (b) captures long-range dependencies from the perspective of structure-level feature, using Transformer-based methods. (c) uses the proposed GAiA-Net to capture long-range dependencies on both pixel-level and structure-level features.

In summary, existing image denoising methods cannot simultaneously capture long-range dependencies on both the pixel-level and structure-level features. Due to this challenge, this paper proposes the **G**raph **A**ttention **i**n **A**ttention **Net**work (GAiA-Net) for image denoising, as shown in Fig.1(c). **Firstly, a novel method of graph construction is introduced for GAiA-Net to learn both the pixel-level and structure-level features.** This method divides the images into patches to formulate the nodes and the edges are initialized using K-nearest neighbors. The data in each node is the pixels in the corresponding patch. Through iterative transformation and learning, different information can be exchanged and aggregated inside (pixel-level) and outside (structure-level) the nodes, respectively. **Secondly, the Graph Attention in Attention (GAiA) is further proposed for GAiA-Net to retrieve long-range dependencies on both pixel-level and structure-level features.** GAiA is designed according to the hierarchical characteristics of the graph data owing to the above graph construction, because the graph of the image is composed of nodes, and nodes contain the data (pixels) within the corresponding patches. Due to this hierarchical property, the proposed GAiA can initially produce pixel-level attention, and then such pixel-level attention is induced to the structure-level feature to compute the final attention on both pixel and structure levels among nodes with various distances. Therefore, the proposed GAiA-Net can capture the long-range dependencies of graph data on both the pixel-level and structure-level features to reconstruct high-quality denoised images. Extensive experiment results demonstrate the proposed GAiA-Net significantly improves the quality of restored images with greatly outperforming the traditional deep denoising methods both quantitatively and visually.

The contributions of this paper can be summarized as follows:

- We propose the **G**raph **A**ttention **i**n **A**ttention **Net**work (GAiA-Net) for image denoising. To our best knowledge, this method is the pioneering attempt to exploit the potential of using long dependencies on both pixel-level and structure-level features for image denoising.
- We introduce a simple yet powerful method for graph construction. Feature maps are divided into patches to form nodes, and the edges are initialized using K-nearest neighbors, with iterative learning enabling information exchange and aggregation within and across nodes, which provides foundation for the proposed GAiA-Net to retrieve different fine-grained features on the pixel and structure levels, respectively.
- We further propose the **G**raph **A**ttention **i**n **A**ttention (GAiA), a hierarchical attention scheme, to establish long dependencies among nodes with various distances on both pixel-level and structure-level features.

The paper is organized as follows: Section II briefly reviews related works on image denoising methods. Sections III presents the proposed GAiA-Net for image denoising. Section

IV demonstrates the experiment results, and the paper is finally concluded in Section V.

II. Related Works

A. CNN-based Image Denoising Method

In the task of image denoising, there are mainly two kinds of approaches, non-learning-based and learning-based methods. Most of the non-learning-based methods reconstruct denoised images by manually designing the prior information to model the noise distribution. These methods include but are not limited to, adaptive prior supplementary models [15], estimated denoising kernel prior models [16], **N**on-local **S**elf-**S**imilar (NSS) models [17], and sparse models [18]. However, the inflexibility of hand-designed prior information limits the development of such conventional non-learning-based image denoising methods. Recently, many excellent CNN-based image denoising models [19], [20], [21] have been proposed, which have greatly improved the performances of image denoising task due to the advent of CNNs. DnCNN (**D**eep **C**onvolutional **N**eural **N**etwork for image denoising) [6] introduces a residual learning method to reconstruct the noise maps, and these noise maps are subtracted from the noisy images to remove the noise. In the discrete wavelet domain, MWCNN (**M**ulti-scale **W**avelet **C**onvolutional **N**eural **N**etwork) [22] introduces a new form of down-sampling and up-sampling layers to build deep networks. Blind noise settings are handled with FFDNet (**F**ast and **F**lexible **D**enoising **N**etwork) [7] by employing a customized noise level map to improve the results. The WINNet (**W**avelet-Inspired **IN**vertible **N**etwork) [23] is proposed to use K-scale lifting inspired invertible neural networks to adaptively adjust soft-thresholds in the LINNs, achieving strong generalization ability and interpretability. Although these CNNs-based models improve the performances of image denoising to some extent, the vanilla convolution layer suffers from the limitation of capturing long-range pixel dependencies.

B. Transformers-based Image Denoising Method

Transformers [24] can capture dependencies over long-range patches using a self-attention mechanism as an alternative to CNNs. IPT (**I**mage **P**rocessing **T**ransformer) [25] is a prior denoising neural network built with standard transformer blocks to reduce image noise. However, IPT requires pre-training on large-scale datasets, and the pre-trained models may limit the further improvement of image denoising performances. Uformer (**U**-shaped Trans**former**) [12] uses a local-enhanced window transformer block and depth-wise convolution to construct a compact network for denoising images. SwinIR (**Swin** Transformer-based **I**mage **R**estoration) [8] hierarchically constructs the image restoration network based on the Swin Transformer blocks with the sliding window operation. Compared with Uformer, SwinIR has a more flexible self-attention mechanism. Restormer (Image **Res**toration with Efficient Trans**former**) [9] reconstructs high-quality images by multi-head attention and feed-forward networks to efficiently capture long-range pixel interactions. Compared with Uformer and SwinIR, Restormer further compresses (compacts) the network structure to use a more flexible attention mechanism for image denoising. SCUNet (**S**win-**C**onv-**U**Net) [26] jointly applies the residual convolution layers and the Swin-Transformer blocks [27] to extract local and non-local features, respectively. The EFF-Net (**E**nhanced **F**requency **F**usion **N**etwork) [28] is proposed to achieve better performance. It addresses the negative impact of tokens with low weight values through the dynamic hash attention module. Additionally, it addresses the missing of textural details through the enhanced frequency fusion module.

C. Graph-based Image Denoising Method

Fortunately, not only Transformer can establish remote pixel dependencies, but also **G**raph **C**onvolution **N**etworks (GCNs) can retrieve dependencies over long-range pixels using Edge-Condition Convolution (ECC), which have been also applied to image denoising. GCDN (**G**raph-**C**onvolutional **D**enoising **N**etwork) [10] treats each pixel in an image as a node and employs ECC to compute the edges among nodes, which overcomes the limitation of vanilla convolution operators for modeling non-local features, and thus improves the image denoising performance. IGNN (**I**mage **G**raph **N**eural **N**etwork) [29] utilizes graph networks to explore the internal recursive properties of image restoration tasks. CPNet (**C**ross-**P**atch **Net**work) [11] captures context relations across k ($k = 3$) neighboring feature patches for better image denoising performance. According to the above demonstrations, Transformer-based and GCNS-based methods retrieve the long dependencies mainly on the structure and pixel levels, respectively. As the diversity and complexity of noise increase, the methods may not be as effective in reducing noise in the image.

III. GAiA-Net

In this section, we first describe the overall structure of **G**raph **A**ttention **i**n **A**ttention **Net**work (GAiA-Net) for image denoising. Then, we provide details of the methods of graph construction and the structure of the basic GAiA block. After that, we introduce GAiA-Net's optimization loss function and implementation details.

A. Overall Pipeline of GAiA-Net

As shown in Fig. 2, the overall structure of the proposed GAiA-Net is an encoder-decoder network composed of projection layers, down-sampling layers, up-sampling layers, and basic GAiA blocks.The input noisy images are first projected by the Input Projection layer. Specifically, given a noisy color image $\mathbf{I}_n \in \mathbb{R}^{H\times W\times 3}$, a Input Projection layer (*i.e.*, 3×3 convolution layer and LeakyReLU) is applied to extract low-level feature maps $\mathbf{F}_l \in \mathbb{R}^{H\times W\times C}$:

$$\mathbf{F}_l = \varphi\left(f_c^{in}\left(\mathbf{I}_n\right)\right), \tag{1}$$

where f_c^{in} denotes the Input Projection layer, and φ represents LeakyReLU activation function. C, H and W denote the numbers of channels, height and width of the noisy image $\mathbf{I}_n$, respectively.

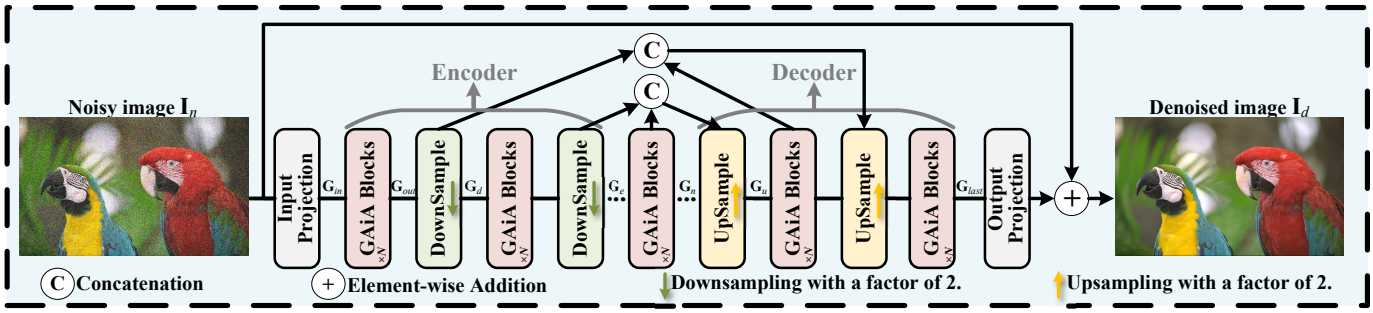

Fig. 2. Overall structure of the proposed GAiA-Net. DownSample and UpSample are down-sampling layer with a factor of 2 and up-sampling layer with a factor of 2, respectively.

Then, the projected feature is transformed into a graph using the proposed method of graph construction. Such input graphs $\mathbf{G}_{in} \in \mathbb{R}^{N\times D}$, where number of patches is N, the D is the dimension of feature vector, are fed to the encoder, which consists of four sets of basic GAiA blocks and down-sampling layers. The GAiA blocks capture the long-range dependencies on both the pixel-level and structure-level features. After each GAiA block in the encoder, a down-sampling layer is attached to extract various graph features of different scales. Meanwhile, since the output produced by the GAiA block is a 1-D graph feature, in the down-sampling layer, the graph feature is first reshaped into a 2-D feature map, which is then down-sampled using a convolution layer with a stride of 2 by doubling the number of channels. Formally, given the graph feature $\mathbf{G}_{out} \in \mathbb{R}^{N\times D}$ output by GAiA block, the down-sampling layer produces the graph feature $\mathbf{G}_d$ as:

$$\mathbf{G}_d = \varphi\left(f_{\downarrow}\left(\mathbf{G}_{out} \underset{\rightarrow}{\mathcal{R}} \mathbb{R}^{H\times W\times C}\right)\right) \underset{\rightarrow}{\mathcal{R}} \mathbb{R}^{2N\times \frac{D}{4}}, \quad (2)$$

where $f_{\downarrow}$ is the down-sampling function with a scaling factor of 2, and $\underset{\rightarrow}{\mathcal{R}}$ is a reshape operation on a tensor. The graph feature generated by the entire encoder is $\mathbf{G}_e \in \mathbb{R}^{16N\times \frac{D}{256}}$. Then, we attach a GAiA block at the end of the encoder as a bottleneck module to further refine the graph feature $\mathbf{G}_n \in \mathbb{R}^{16N\times \frac{D}{256}}$.

Similarly, the produced graph features with different scales from the encoder are fed to the decoder. The decoder is composed of four groups of up-sampling layers and GAiA blocks. Through the up-sampling layer, the decoder gradually recovers the high-resolution graph feature from the low-resolution graph feature $\mathbf{G}_n \in \mathbb{R}^{16N\times \frac{D}{256}}$. In the up-sampling layer, we use deconvolution operation with the stride of 2 and the kernel size of 2 × 2 for upsampling, while reducing the channels and increasing the spatial resolution of feature maps. The up-sampling layer produces the graph feature $\mathbf{G}_u$ as:

$$\mathbf{G}_u = \varphi\left(f_{\uparrow}\left(\mathbf{G}_n \underset{\rightarrow}{\mathcal{R}} \mathbb{R}^{\frac{H}{16}\times\frac{W}{16}\times 16C}\right)\right) \underset{\rightarrow}{\mathcal{R}} \mathbb{R}^{8N\times\frac{D}{64}}, \quad (3)$$

where $f_{\uparrow}$ is the up-sampling function with a scaling factor of 2. To enhance the multi-scale graph feature to reconstruct denoised images, the skip connection is employed to connect the graph features generated by the GAiA blocks in the encoder and by the up-sampling layer in the decoder at the node dimension. The concatenated graph feature is fed to the subsequent GAiA blocks in the decoder.

Finally, based on Eqn. 4, a denoised image $\mathbf{I}_d \in \mathbb{R}^{H\times W\times 3}$ is reconstructed from the final graph feature produced by the decoder using an Output Projection layer (*i.e.*, a convolution layer with a kernel size of 3 × 3).

$$\mathbf{I}_d = f_c^{out}\left(\mathbf{G}_{last} \underset{\rightarrow}{\mathcal{R}} \mathbb{R}^{H\times W\times C}\right) + \mathbf{I}_n, \quad (4)$$

where f_c^{out} denotes the convolution function in the Output Projection layer, and $\mathbf{G}_{last}$ is the aggregation of graph feature.

B. Graph Construction

The low-level projected feature maps $\mathbf{F}_l \in \mathbb{R}^{H\times W\times C}$ are first divided into N patches. We get $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_i, \cdots, \mathbf{x}_N]$ by transforming each feature patch into a feature vector $\mathbf{x}_i \in \mathbb{R}^D$ $(i = 1, 2, \cdots, N)$, where D is the dimension of feature vector. The feature vectors $\mathbf{X}$ are considered to be an unordered set of nodes. For each node $\mathbf{x}_i$, $\mathcal{K}(\mathbf{x}_i)$ is its k nearest neighbors. The edge $\mathbf{e}_{ji} \in \mathbf{E}$ is added from $\mathbf{x}_j$ to $\mathbf{x}_i$ for all $\mathbf{x}_j \in \mathcal{K}(\mathbf{x}_i)$, where $\mathbf{E}$ denotes all the edges. Thus, a graph, denoted as $\mathbf{G} = (\mathbf{X}, \mathbf{E})$, can be constructed. Within the framework of this hierarchical construction, there are distinct modes of information exchange and aggregation that take place both within the nodes themselves, at the pixel-level feature, and beyond the nodes, at the structure-level feature. Hence, through iterative transformation and learning, the proposed graph construction can retrieve features on both the pixel and structure levels. The process of exchange of pixel-level information transpires within the nodes. The aggregation and exchange of information occur among these nodes, with edges linking those that are proximate in the feature domain. Consequently, the graph is capable of encapsulating the associations between pixel-specific characteristics, thereby promoting the extraction of these features via repetitive adaptation and learning.

C. GAiA Block

The proposed GAiA-Net is constructed by the basic GAiA blocks (Fig. 3), which are sequentially composed of two stages, *i.e.*, Attentive Stage and Refinement Stage.

Attentive Stage. The Attentive Stage aims to capture long-range dependencies from both the pixel-level (*i.e.*, internal information of the node) and structure-level (*i.e.*, information

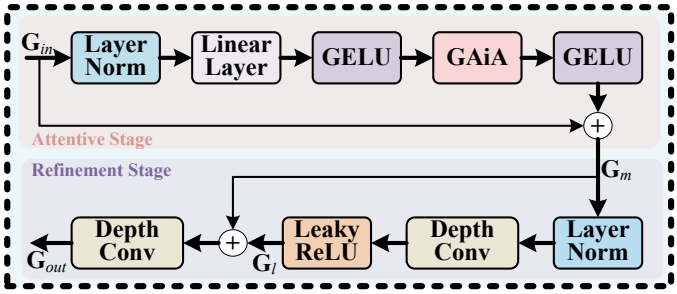

Fig. 3. Structure of the proposed GAiA block. It contains Attentive Stage and Refinement Stage, respectively. Attentive Stage aims to capture long-range dependencies from both the pixel-level and structural-level features. Refinement Stage is to further enhance the contextual information. Compared to ReLU, GELU [30] has a smoother function curve and faster convergence speed.

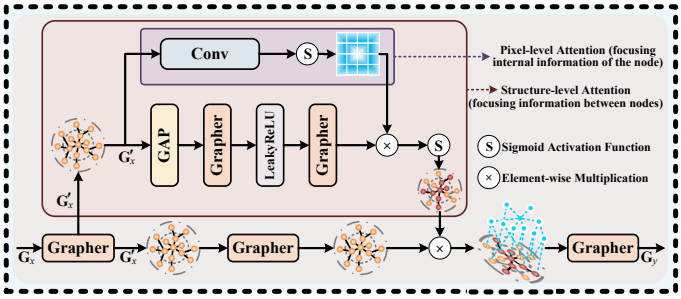

Fig. 4. Structure of the **G**raph **A**ttention **i**n **A**ttention (GAiA) mechanism. Grapher and GAP are graph convolution operation and global average pooling function, respectively.

among nodes with various distances) features. Since the proposed method of graph construction method establishes the graph hierarchically, to keep such hierarchical consistency in the graph transformation and learning, we propose the GAiA mechanism, which is the core component of the Attentive Stage. The detailed structure of the proposed GAiA is shown in Fig. 4. The GAiA first generates the pixel-level attention to be induced to the structure-level feature to produce the final attention. Therefore, the GAiA can jointly use both the pixel and structure information to hierarchically produce the compound long-dependencies, which can more sufficiently retrieve the complex noise information than traditional methods.

Specifically, given the input graph feature $\mathbf{G}_x \in \mathbb{R}^{N\times D}$, the graph convolution operation $\mathcal{F}$ [31] can be formulated as the following aggregation and update operations:

$$\begin{aligned}\mathbf{G}_x^{'} &= \mathcal{F}\left(\mathbf{G}_x, \mathbf{W}\right) \\ &= Update\left(Aggregate\left(\mathbf{G}_x, \mathbf{W}_{agg}\right), \mathbf{W}_{update}\right),\end{aligned} \tag{5}$$

where $\mathbf{G}_x^{'}$ is the output graphs, $\mathbf{W}_{agg}$ and $\mathbf{W}_{update}$ are the learnable weights of the aggregation and update functions, respectively. In the $Aggregate$ aggregation operation, features from neighboring nodes are aggregated to represent a node:

$$\mathbf{x}_i' = h\left(\mathbf{x}_i, g\left(\mathbf{x}_i, \mathcal{X}\left(\mathbf{x}_i\right), \mathbf{W}_{agg}\right), \mathbf{W}_{update}\right), \tag{6}$$

where $\mathcal{X}\left(\mathbf{x}_i\right)$ is the set of neighbor nodes of $\mathbf{x}_i$. Because max-relative graph convolution [32] is simple and efficient, we adopt it here:

$$\begin{aligned} g(\cdot) &= \mathbf{x}_i'' = \max\left(\{\mathbf{x}_i - \mathbf{x}_j \mid j \in \mathcal{X}\left(\mathbf{x}_i\right)\}\right), \\ h(\cdot) &= \mathbf{x}_i' = \mathbf{x}_i''\mathbf{W}_{\text{update}}, \end{aligned} \tag{7}$$

where the bias term is omitted. $g(\cdot)$ and $h(\cdot)$ are the functions used in aggregation and update operations.

To produce the pixel-level attention within node, $\mathbf{G}_x^{'}$ is first projected by a convolution layer f_{cp} with kernel size of 1×1 and a Sigmoid activation layer φ_s. This process is formulated as follows:

$$\mathbf{A}_p = \varphi_s\left(f_{cp}\left(\mathbf{G}_x^{'}\right)\right). \tag{8}$$

where $\mathbf{A}p$ is pixel-level attention. Then, to produce the structure-level feature, $\mathbf{G}_x^{'}$ is first processed by a Global Average Pooling (GAP) layer to model the information among nodes in the graph feature, which can be formulated as follows:

$$\boldsymbol{m} = \frac{1}{D}\sum \mathbf{G}_x^{'}, \tag{9}$$

where $\boldsymbol{m} \in \mathbb{R}^{N\times 1}$. Such averaged graph feature $\boldsymbol{m}$ is sequentially projected by the graph convolution, LeakyReLU activation, and graph convolution. This operation set is denoted as $\mathcal{F}_{GLG}$, producing the structure-level feature $\mathcal{F}_{GLG}(\boldsymbol{m})$. Hence, through inducing the pixel-level attention $\mathbf{A}_p$ into the structure-level feature $\mathcal{F}_{GLG}(\boldsymbol{m})$, the final attention is formulated as follows:

$$\mathbf{A}_f = \varphi_s(\mathbf{A}_p \otimes \mathcal{F}_{GLG}\left(\boldsymbol{m}\right)), \tag{10}$$

where $\otimes$ denotes the standard element-wise multiplication. The term $\mathbf{A}_p \otimes \mathcal{F}_{GLG}\left(\boldsymbol{m}\right)$ denotes the attentive structure feature among all nodes within graph. From the above equation, the pixel-level attention emphasizes the nodes, which are more correlated to the internal information of node, on the structure-level feature.

The final attention on both the pixel and structure levels will emphasize the significant elements of its input graph feature $\mathbf{G}_x$, shown as below:

$$\mathbf{G}_y = \mathbf{A}_f \otimes \mathcal{F}\left(\mathbf{G}_x^{'}\right), \tag{11}$$

where $\mathbf{G}_y$ denotes the output graph feature of the proposed GAiA mechanism.

Refinement Stage. Actually, neighbor pixels are crucial references for image denoising [33]. Therefore, to further utilize the contextual information inside nodes, we propose Refinement Stage. This stage aims to enhance the contextual information inside nodes within graph feature. As shown in Fig. 3, we first apply a Layer Normalization function to graph feature $\mathbf{G}_m \in \mathbb{R}^{N\times D}$ to compute the horizontal normalization of different nodes. Next, we reshape the graph features (i.e. data within all nodes) into a 2D feature map and use a 3×3 depth-wise convolution to capture the contextual information $\mathbf{G}_l$ (local pixel-level information) of the feature maps, as shown in Eqn. 12:

$$\mathbf{G}_l = \varphi\left(f_d\left(\mathrm{LN}\left(\mathbf{G}_m\right) \underset{\rightarrow}{\mathcal{R}} \mathbb{R}^{H\times W\times C}\right)\right), \tag{12}$$

where LN denotes the Layer Normalization function, f_d is a 3×3 depth-wise convolution function. Finally, we further

refine such feature through another depth-wise convolution layer, as shown in Eqn 13:

$$\mathbf{G}_{out} = f_d\left(\mathbf{G}_l + \left(\mathbf{G}_m \underset{\nearrow}{\mathcal{R}} \mathbb{R}^{H\times W\times C}\right)\right) \underset{\nearrow}{\mathcal{R}} \mathbb{R}^{N\times D}, \quad (13)$$

where $\mathbf{G}_{out}$ is the output feature of the GAiA block.

D. Loss Function

As the minimum absolute error loss (*i.e.*, $\mathcal{L}_1 = \frac{1}{N_t}\sum|\mathbf{I}_d - \mathbf{I}_{traget}|$ [34]) can not be differentiated at zero, the reconstructed denoised image is unstable. Similarly, as a result of mean squared error loss (*i.e.*, $\mathcal{L}_2 = \frac{1}{N_t}\sum(\mathbf{I}_d - \mathbf{I}_{traget})^2$ [35]), the denoised images may lack perceptual realism and be too smooth, see section IV-C for details. Hence, for the proposed GAiA-Net, we adopt Charbonnier loss [26] with the following equation:

$$\mathcal{L} = \frac{1}{N_t}\sum\sqrt{\|\mathbf{I}_d - \mathbf{I}_{traget}\|^2 + \epsilon^2}. \quad (14)$$

where N_t denotes the number of training samples, $\mathbf{I}_d$ denotes the denoised image produced from the proposed GAiA-Net, and $\mathbf{I}_{traget}$ represents the ground-truth corresponding to the input noise image, and ϵ^2 is a constant that is empirically set to 1×10^{-6}.

IV. Experiments

In this section, we first demonstrate the implementation and initialization of the proposed GAiA-Net. Next, we compare the performance of the proposed GAiA-Net with state-of-the-art image denoising methods on various synthetic noisy image datasets and real noisy image datasets. Finally, an ablation study on the GAiA-Net is conducted to verify the effectiveness of the proposed GAiA-Net.

A. Implementation and Initialization

The proposed GAiA-Net consists of an Input Projection layer, an Output Projection layer, an encoder, and a decoder. The encoder is composed of 4 groups of GAiA blocks and down-sampling layers, and the decoder is composed of 4 groups of GAiA blocks and up-sampling layers. Both the down-sampling and up-sampling scale factors are set to 2. The kernel size of the convolution layers in the down-sampling and up-sampling layers are set to 6×6. Each input training batch of the proposed GAiA-Net contains 128 patches with the size of 256×256 from the pairs of training images. We use the Adam [36] optimizer to train our GAiA-Net with setting β_1 and β_2 to 0.9 and 0.999, respectively. The learning rate is set to 1×10^{-4} in the training process.

B. Experiment Comparisons

A comparison of the proposed method with many state-of-the-art image denoising methods is performed to verify the advanced performance of the proposed GAiA-Net. We report their results using publicly available implementations provided by the corresponding literature to make a fair comparison. In this section, three types of noisy image datasets, *i.e.*, synthetic noise image datasets, real noise image datasets, and real Poisson-Gaussian biomedical noise image datasets, are used to compare and evaluate the proposed GAiA-Net.

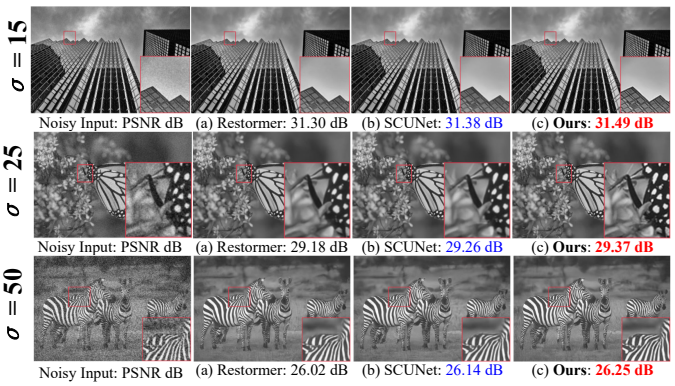

Fig. 5. Visual comparisons between our GAiA-Net and its competitors in the evaluation of gray-scale noisy image denoising. The test images are selected from BSD68, Set12 and Urban100 with different noisy levels of σ=15, 25, 50. **Red** and **blue** are the best and second best objective indicators, respectively.

TABLE I
Average PSNRs of the denoised gray-scale images from Set12, BSD68 and Urban100 datasets. The values of PSNRs are positively correlated with visual quality.

Dataset	Set12				BSD68				Urban100			
Methods	σ=15	σ=25	σ=50	σ=75	σ=15	σ=25	σ=50	σ=75	σ=15	σ=25	σ=50	σ=75
DnCNN	32.86	30.44	27.18	25.93	31.73	29.23	26.23	25.41	32.64	29.95	26.23	24.87
IRCNN	32.77	30.38	27.14	N/A	31.63	29.15	26.19	N/A	32.46	29.80	26.22	N/A
FFDNet	32.75	30.43	27.32	26.32	31.63	29.19	26.29	25.55	32.40	29.90	26.50	25.37
MWCNN	33.15	30.79	27.74	N/A	31.88	29.41	26.53	N/A	33.17	30.66	27.42	N/A
NLRN	33.16	30.80	21.64	N/A	31.88	29.41	26.47	N/A	33.45	30.94	21.49	N/A
RNAN	N/A	N/A	27.70	N/A	N/A	N/A	26.48	N/A	N/A	N/A	27.65	N/A
FOCNet	33.07	30.73	27.68	N/A	31.83	29.38	26.50	N/A	33.15	30.64	27.40	N/A
DAGL	33.28	30.93	27.81	N/A	31.93	29.46	26.51	N/A	33.79	31.39	27.97	N/A
DRUNet	33.25	30.40	29.90	N/A	31.91	29.48	26.59	N/A	33.40	31.11	27.96	N/A
SwinIR	33.36	31.01	27.91	N/A	31.97	29.50	26.58	N/A	33.70	31.30	27.98	N/A
Restormer	33.42	31.08	28.00	N/A	31.96	29.52	26.62	N/A	33.79	31.46	28.29	N/A
EFF-Net-T	33.43	31.09	28.04	27.49	31.99	29.55	26.67	26.15	33.88	31.58	28.56	27.98
GAiA-Net (Ours)	**33.54**	**31.20**	28.18	**27.65**	**32.09**	**29.67**	**26.75**	**26.18**	**33.92**	**31.68**	**28.70**	**28.16**

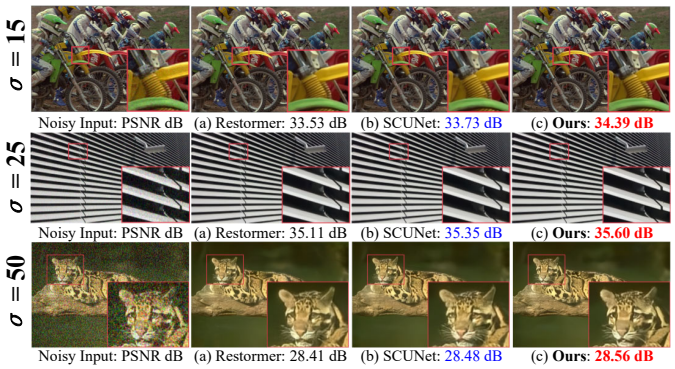

Fig. 6. Visual comparisons between our GAiA-Net and its competitors in the evaluation of color noisy image denoising. The test images are selected from CBSD68, Kodak24 and McMaster with different noisy levels of σ=15, 25, 50. **Red** and **blue** are the best and second best objective indicators, respectively.

Synthetic Gray-scale Noisy Images. We evaluate the image denoising performance on gray-scale image datasets (Set12 [6], BSD68 [37], and Urban100 [38]) with different levels of noise (σ = 15, 25, 50). Table I reports the results of denoising synthetic noisy images on such three gray-scale image test datasets. The proposed GAiA-Net is compared with twelve state-of-the-art denoising methods, including DnCNN [6], IRCNN [39], FFDNet [7], MWCNN [22], NLRN [40], RNAN [41], FOCNet [42], DAGL [13], DRUNet [43], SwinIR

TABLE II
Average PSNRs of the denoised color images from CBSD68, Kodak24, McMaster and Urban100 datasets. The values of PSNRs are positively correlated with visual quality.

Dataset	CBSD68			Kodak24			MCMaster			Urban100		
Methods	σ=15	σ=25	σ=50	σ=15	σ=25	σ=50	σ=15	σ=25	σ=50	σ=15	σ=25	σ=50
EM3D	33.52	30.71	27.38	34.28	32.15	28.46	34.06	31.66	28.51	33.93	31.36	27.93
DnCNN	33.90	31.24	27.95	34.60	32.14	28.95	33.45	31.52	28.62	32.98	30.81	27.59
IRCNN	33.86	31.16	27.86	34.69	32.18	28.93	34.58	32.18	28.91	33.78	31.20	27.70
EFDNet	33.87	31.21	27.96	34.63	32.13	28.98	34.66	32.35	29.18	33.83	31.40	28.05
BRDNet	34.10	31.43	28.16	34.88	32.41	29.22	35.08	32.75	29.52	34.42	31.99	28.56
DRUNet	34.30	31.69	28.51	35.31	32.89	29.86	35.40	33.14	30.08	34.81	32.60	29.61
SwinIR	34.42	31.78	28.56	35.34	32.89	29.79	35.61	33.20	30.22	35.13	32.90	29.82
Restormer	34.40	31.79	28.60	35.47	33.04	30.01	35.61	33.34	30.30	35.13	32.96	30.02
EDT-B	34.39	31.76	28.56	35.37	32.94	29.87	35.61	33.34	30.25	35.22	33.07	30.16
SCUNet	34.40	31.79	28.61	35.34	32.92	29.87	35.60	33.34	30.29	35.18	33.03	30.14
EFF-Net-T	34.39	31.75	28.55	35.30	32.91	29.93	35.51	33.33	30.22	35.11	33.01	30.05
GAiA-Net (Ours)	**34.49**	**31.82**	**28.73**	**35.51**	**32.99**	**30.08**	**35.68**	**33.41**	**30.33**	**35.29**	**33.11**	**30.20**

[8], Restormer [9], EDT-B [44], SCUNet [26], and EFF-Net-T [28]. On all the three gray-scale image datasets with different noise levels (σ = 15, 25, 50), the proposed GAiA-Net achieves significantly better PSNR results than all the other twelve state-of-the-art methods. Specifically, on different noise levels, PSNRs gained by our GAiA-Net are all higher than those obtained by the comparison methods. Moreover, compared to the best SCUNet, GAiA-Net averagely achieves 0.12 dB, 0.10 dB, and 0.10 dB improvements of PSNRs on Set12, BSD68, and Urban100, respectively. This indicates the effectiveness of the proposed GAiA-Net for denoising synthetic noisy gray-scale images. Furthermore, we conducted additional experiments at noise levels of $\sigma = 75$ and compared the performance of our GAiA-Net with state-of-the-art methods. Specifically, the proposed GAiA-Net achieved an average PSNR of 27.33 dB, while the closest competing method (SCUNet) achieved an average PSNR of 27.21 dB. Our experimental results indicate that the proposed GAiA-Net still outperforms other methods in terms of image denoising performance, even at such high noise levels.

Additionally, more visual comparisons on different gray-scale image datasets with σ = 15, 25, 50 are shown in Fig. 5. In comparison with the other twelve state-of-the-art methods, our method can recover images from synthetic noisy images more satisfactorily without apparent over-smoothness or artifacts. For example, building window outlines, butterfly antennae, and stripes on zebras reconstructed with GAiA-Net are more clearly visible than other comparison methods. This indicates that the compared methods have a severe bottleneck in reconstruction of texture and contour details. However, the proposed GAiA-Net using the novel graph construction can lay the foundation for the GAiA block to capture long-range dependencies at the perspectives of pixel-level and structure-level features to produce more satisfactory visualization results.

Synthetic color noisy images. Table II reports the results of denoising synthetic color noisy images on the CBSD68 [37], Kodak24 [45], McMaster [46], and Urban100 [38] datasets. The proposed GAiA-Net is compared to ten state-of-the-art denoising methods, including BM3D [18], DnCNN [6], IRCNN [39], FFDNet [7], BRDNet [47], DRUNet [43], SwinIR [8], Restormer [9], EDT-B [44], SCUNet [26], and EFF-Net-T [28]. It can be seen that the proposed GAiA-Net achieves best PSNRs than all the other state-of-the-art methods for all the noise levels on all the four color image datasets. Specifically, the proposed GAiA-Net exceeds SCUNet by an average PSNR of 0.08dB on CBSD68, 0.12dB on Kodak24, 0.06dB on McMaster, and 0.08dB on Urban100. This shows that our GAiA-Net can effectively remove the noise for the synthetic color noisy images.

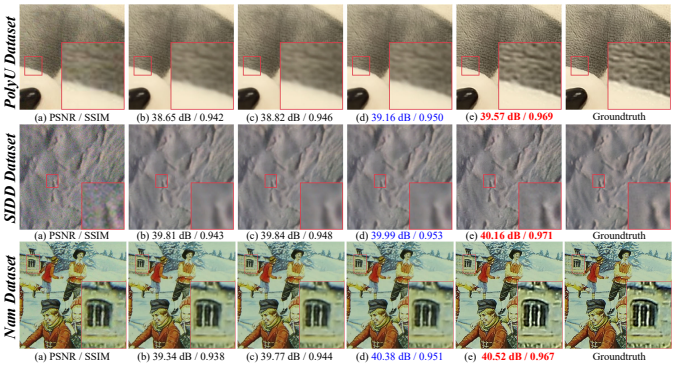

Fig. 7. Visual comparisons between the proposed method and its competitors in the evaluation of real noisy image denoising. In the example row of the PolyU dataset, (a) noisy image, (b) VDN (c) AINDNet, (d) MIRNet, and (e) GAiA-Net; In the example row of the SIDD dataset, (a) noisy image, (b) APD-Net (c) Uformer, (d) Restormer, and (e) GAiA-Net; In the example row of the Nam dataset, (a) noisy image, (b) AINDNet (c) MIRNet, (d) APD-Net, and (e) GAiA-Net. **Red** and **blue** are the best and second best objective indicators, respectively.

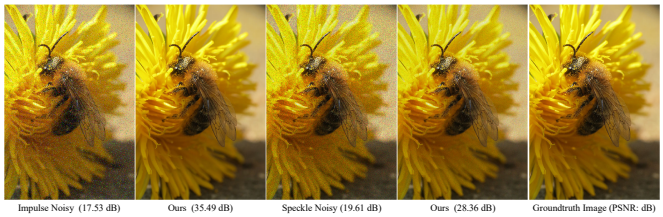

Fig. 8. Evaluate the visualization of GAiA-Net using pulse noise and speckle noise with a noise intensity of 5%.

To qualitatively evaluate the proposed GAiA-Net, Fig. 6 shows the visual comparisons of different methods on different synthetic color noisy image datasets with $\sigma = 15, 25, 50$. All the compared methods can remove the noise in the color noisy images to some extent, while easily leading to over-smoothing and artifacts. Although SCUNet and Restormer achieve a larger PSNR gain, they smear many texture regions in the reconstructed denoised images, which indicates that the above methods produce an unsatisfactory performance in non-local reconstruction. On the contrary, the proposed GAiA-Net produces more satisfactory visualization results without generating other image artifacts. For example, the pattern at the end of the whiskers and on the forehead of the leopard's face reconstructed by GAiA-Net is more clearly visible than the other comparison methods. The possible reason is that GAiA-Net can capture the long-range dependencies on both the pixel-level and structure-level features to effectively improve the visualization of the denoised images.

Real-World Noisy Images. In this section, we evaluate the performance of the proposed GAiA-Net on real-world noisy images with complex and unknown noise statistics. The real noise contained in real-world images is usually accumulated from multiple complex noise sources. Therefore, evaluating

TABLE III
Average PSNRs of the denoised real noisy images from Nam, PolyU and SIDD datasets. The values of PSNRs and SSIMs are positively correlated with visual quality.

Dataset	SIDD		Nam		PolyU	
Methods	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
DnCNN-B	38.56	0.910	36.08	0.903	35.74	0.878
FFDNet	38.60	0.909	37.85	0.938	37.19	0.939
TWSC	35.89	0.838	38.37	0.952	37.63	0.954
CBDNet	38.68	0.909	38.51	0.957	37.85	0.956
RIDNet	38.71	0.913	38.72	0.960	38.07	0.957
VDN	39.29	0.911	39.16	0.965	38.43	0.960
GCDN	38.93	0.910	38.96	0.962	38.21	0.958
PAN-Net	39.33	0.912	40.18	0.978	39.91	0.971
AINDNet	39.45	0.915	39.21	0.966	38.78	0.963
APD-Nets	39.75	0.959	40.36	0.989	N/A	N/A
MIRNet	39.71	0.959	39.88	0.973	39.25	0.971
HPDNet	39.72	0.958	40.26	0.979	39.89	0.970
Uformer	39.77	0.959	N/A	N/A	N/A	N/A
Restormer	40.02	0.960	N/A	N/A	N/A	N/A
EFF-Net	39.79	0.957	39.98	0.974	39.87	0.968
GAiA-Net (Ours)	**40.09**	**0.961**	**40.40**	**0.989**	**39.99**	**0.972**

the denoising performance of the proposed method on real noisy images is of great value for practical applications.

Table III reports the results of denoising real noisy images on the SIDD [48], PolyU [49], and Nam [50] datasets. The proposed GAiA-Net is compared with fourteen state-of-the-art denoising methods. The compared methods include DnCNN [6], IRCNN [39], FFDNet+ [7], TWSC [51], CBDNet [52], RIDNet [19], VDN [53], PAN-Net [54], AINDNet [55], APD-Nets [34], Uformer [12], Restormer [9], MIRNet [56], HPDNet [57], and EFF-Net [28]. It can be seen that the proposed GAiA-Net largely increases the PSNR/SSIM results more than the other fourteen state-of-the-art methods on all three real noisy image datasets. For the Nam dataset, GAiA-Net exceeds APD-Nets by an average PSNR of 0.04dB. This demonstrates that the proposed GAiA-Net structure is effective for denoising real noisy images. This further implies that GAiA-Net utilizing the long-range dependencies from both the pixel-level and structure-level features can more effectively remove the complex real-world noise, producing the best image denoising performance among all the compared methods.

To visually demonstrate the superiority of our method, Fig. 7 shows a visual comparison of different methods on different datasets for denoising real noisy images. We can observe that our GAiA-Net achieves the best visual results in terms of noise removal and detail preservation. For example, neither Uformer nor Restormer can preserve the granularity of toy bear fur, while the proposed GAiA-Net can well reconstruct and preserve fine hairs. In addition, compared to other methods, the SSIM indicator of the proposed GAiA-Net is the best, indicating that our proposed method effectively preserves edges. This further demonstrates that retrieving long-range dependencies from both the pixel-level and structure-level features can effectively remove the complex noise by recovering the better details to significantly improve the image denoising performance quantitatively and qualitatively.

Synthetic Impulse and Speckle Noisy Images. To validate the effectiveness of the proposed method, we evaluated the image denoising performance of GAiA-Net using impulse noise and speckle noise with a noise intensity of 5%. As shown in Fig. 8, the proposed GAiA-Net can effectively remove impulse noise and speckle noise, especially in the case of removing impulse noise. This indicates that GAiA-Net can effectively capture the long-range dependencies of both pixel-level and structure-level features, enabling it to remove different types of noise and recover high-quality denoised images.

TABLE IV
Average PSNRs of the denoised real Poisson-Gaussian biomedical noise images from FMD datasets. The values of PSNRs are positively correlated with visual quality.

Methods	Number of raw images for averaging				
	1	2	4	8	16
DnCNN	34.88	36.02	37.57	39.28	41.57
DRUNet	34.96	36.13	37.60	39.28	41.60
SwinIR	34.99	36.26	37.76	39.38	41.79
Restormer	35.04	36.34	37.84	39.41	41.86
SCUNet	35.17	36.49	37.90	39.46	41.91
GAiA-Net (Ours)	**35.51**	**36.63**	**37.94**	**39.58**	**42.05**

Real Poisson-Gaussian Biomedical Noise Images. We evaluate the denoising performance of the proposed GAiA-Net on the biomedical noisy image dataset (*i.e.*, FMD [58] is employed for comparisons of real Poisson-Gaussian biomedical noise images), which contains the unknown noise intensity generated during the imaging of biomedical images under a microscope. Although the biomedical image registration under the microscope is of high quality, it still needs to repeat an average of 50 captures to obtain reliable ground truth. Therefore, evaluating the denoising performance of this method on real Poisson-Gaussian biomedical noise images is of great value for practical imaging applications.

Table IV reports the results of denoising real Poisson-Gaussian biomedical noise images on the FMD datasets. The proposed GAiA-Net is compared with five state-of-the-art denoising methods. The compared methods include DnCNN [6], DRUNet [43], SwinIR [8], Restormer [9], and SCUNet [26]. On real Poisson-Gaussian noisy image datasets, GAiA-Net significantly increases the PSNR results compared to the other five state-of-the-art methods. Specifically, when using the average of 1 raw image as the input image, GAiA-Net exceeds SCUNet by an average PSNR of 0.34 dB. Obviously, the proposed GAiA-Net structure is shown to be effective for denoising real Poisson-Gaussian noisy images in this study. This is mainly because the long-range dependencies of both the pixel-level and structure-level features can remove complex real Poisson-Gaussian noise more effectively than any other compared method.

In Fig. 9, a visual comparison of different methods on FMD datasets for denoising real Poisson-Gaussian noisy images are shown to demonstrate the superiority of the proposed GAiA-Net. Regarding noise removal, object structure edges, and details preservation, we observe that our GAiA-Net results are the best. The proposed GAiA-Net, for instance, can well reconstruct and preserve finer cell edges and details, whereas

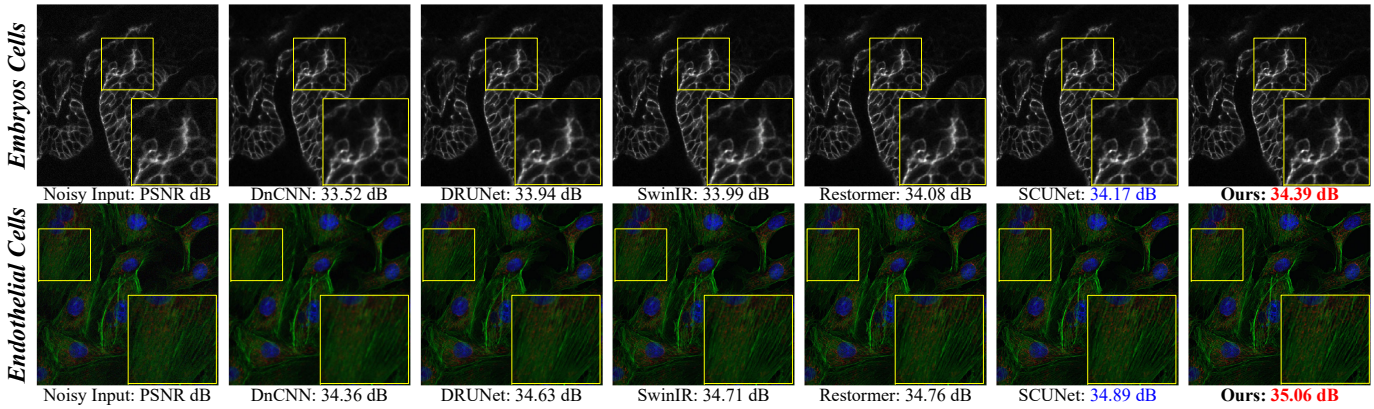

Fig. 9. Visual comparisons between the proposed method and its competitors in the evaluation of real Poisson-Gaussian noisy image denoising. **Red** and **blue** are the best and second best objective indicators, respectively. Best viewed by zooming.

TABLE V
Inference time, FLOPs, and #Param. comparisons on image sizes of 256 × 256 on an Nvidia RTX Titan GPU with state-of-the-art methods. The results from all the compared methods in this table are obtained by inferring an image with the size of 256 × 256 on the same GPU device (*i.e.*, an Nvidia RTX Titan GPU).

Metrics	DRUNet	MIRNet	MPRNet	APD-Nets	SwinIR	Restormer	GAiA-Net
#Param.	32.64M	31.78 M	15.74 M	18.61M	11.49M	28.13M	27.57M
FLOPs	143.5G	555.5G	588.1G	283.25 G	787.9G	140.1G	20.7G
Inference time	0.020s	2.082s	1.058s	0.853s	0.525s	0.081s	0.079s

TABLE VI
Effect of the number of k-nearest neighbors on the performance of GAiA-Net for image denoising.

k-nearest Number	1	2	3	4	5	6	7	8	9	10
CBSD68	34.20	34.28	34.36	**34.49**	34.44	34.38	34.33	34.28	34.22	34.17
SIDD	39.37	39.61	39.85	**40.09**	39.93	39.78	39.62	39.47	39.31	39.15

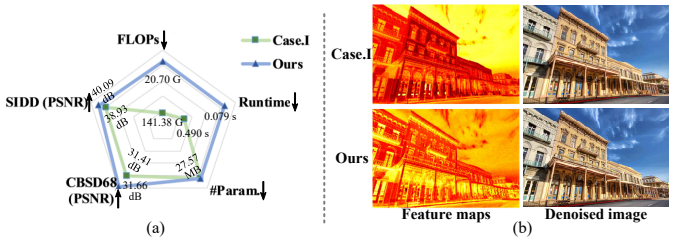

Fig. 10. Comparisons of PSNR value and the parameters on the real noisy image testing sets. (a) Performance effect of our graph construction on image denoising. (b) Effects of our graph construction on reconstructing denoised images and capturing feature maps. The visualized average feature map is taken from the last GAiA block of the proposed GAiA-Net.

neither Restormer nor SCUNet can preserve such granularity. This further demonstrates that our method of retrieving long-range dependencies from both pixel-level and structure-level features can effectively improve the denoised images quantitatively and qualitatively.

Efficiency Comparison. An essential benchmark for evaluating the practicality of image denoising methods is their ability to effectively remove image noise. This section focuses on comparing GAiA-Net, a novel image denoising method, with the latest state-of-the-art approaches in terms of their denoising efficiency. To ensure fair comparisons of efficiency, we employ FLOPs, inference time, and parameter count as metrics. Specifically, we conduct the comparisons on the same computer equipment (namely, one equipped with an Nvidia RTX Titan GPU) for accuracy, as presented in Table V.

The data presented in this table reflects results obtained from testing image inference on a 256 × 256 image size, using the same GPU device across all compared methods. The proposed GAiA-Net exhibits the lowest FLOPs, in stark contrast to Restormer which, despite its self-attention mechanism, yields high FLOPs and long inference times. Moreover, the intricate network structures of MIRNet and APD-Net also lead to high FLOPs and long inference times. In comparison, GAiA-Net strikes an optimal balance between FLOPs and inference time, making it a standout performer in both efficiency and denoising capabilities. Indeed, in terms of both performance and efficiency, our proposed GAiA-Net boasts significant advantages over other state-of-the-art denoisers.

C. Ablation Study

To visually illustrate the impacts of the proposed novel graph construction and GAiA block on image denoising, we use the following cases to conduct experiments. Case.I means that the proposed graph construction is not used in the GAiA-Net structure, *i.e.*, each pixel in the image is treated as a node. Case.II denotes that only the graph convolution operator (*i.e.*, replacing all the GAiA blocks) is used in the GAiA-Net structure. Case.III represents that the proposed GAiA block without pixel-level attention is used in the GAiA-Net structure. Except for the above variations, the experimental environment, experimental settings, and overall network structure of all the cases are consistent.

Effect of Our Graph Construction. To quantify the impact of our graph construction on image denoising performance, the main evaluation metrics (*e.g.*, PSNR, FLOPs, Runtime, and #Param.) of Case.I, and the proposed GAiA-Net on synthetic color image dataset (*i.e.*, CBSD68) and real image dataset (*i.e.*, SIDD) are reported in Fig. 10 (a), respectively. Compared with Case.I employing the each pixel in the image as a node, GAiA-

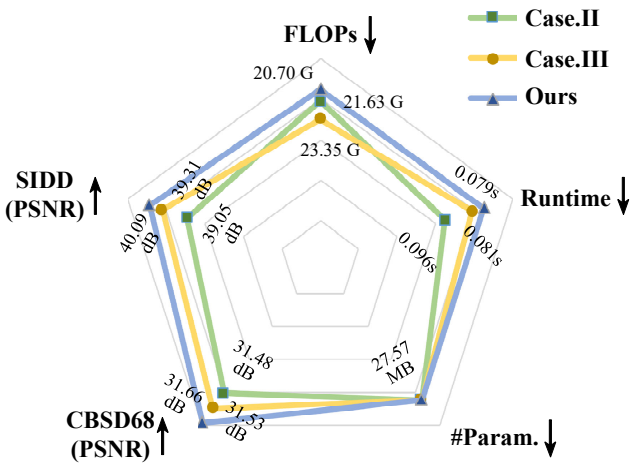

Fig. 11. Performance effect of the proposed GAiA block on image denoising. The PSNR, FLOPs, Runtime, and #Param. are used as reference metrics for image denoising performance on the CBSD68 and SIDD datasets.

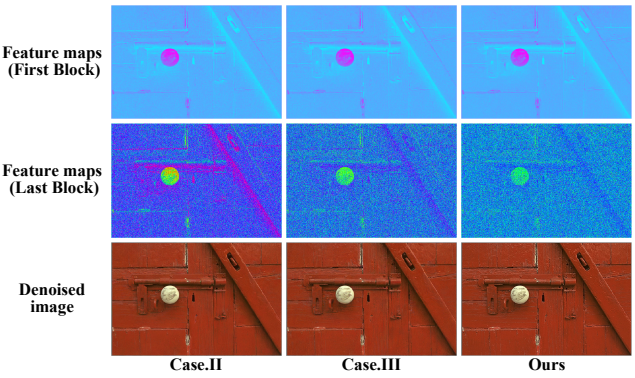

Fig. 12. Effect of our GAiA block on reconstructing denoised images and capturing feature maps.The GAiA block generates denser noise feature maps and better denoised images than other methods by using attention to capture long-range dependencies from pixel-level and structure-level features.

Net improves PSNR by **0.25 dB** and **1.16 dB** on CBSD68 and SSID datasets, respectively. This illustrates that our graph construction can improve the image denoising performance by exchanging and aggregating different information inside (pixel-level) and outside (structure-level) the nodes, respectively. Furthermore, we also compare the FLOPs, Runtime, and the number of parameters (#Param.) to measure the comprehensive performances of image denoising methods. To fairly evaluate the methods, the results from all the compared methods in this figure are obtained by inferring an image with the size of 256×256 on the same GPU device. In Fig. 10 (a), it is worth noting that, compared to the Case.I, the proposed GAiA-Net largely reduces the FLOPs and Runtime by about **6.83** times and **6.20** times.

As shown in Fig. 10 (b), compared to Case.I applying each pixel in the image as a node, in terms of visualizing feature maps, our GAiA-Net retains and enhances more structure information (*e.g.*, white clouds with edges). Moreover, such a recovered structure can further guide the proposed GAiA-Net to produce better details, for example, the finer texture of the white clouds. Therefore, this is the main reason why the feature maps in Case.I and the proposed GAiA-Net have large

TABLE VII
Effect of different loss functions on the performance of reconstructing denoised images.

Loss Function	MAE (L1) Loss	MSE (L2) Loss	Charbonnier Loss
CBSD68	34.13	34.22	**34.36**
SIDD	39.61	39.74	**39.85**

differences for the same input images. This fully demonstrates that, our graph construction can effectively and adequately retrieve both the structure and pixel information by the established hierarchical graph through the iterative transformation and learning to improve the feature representation ability.

Effect of the Number of k-nearest Neighbors on the GAiA-Net. As shown in Table VI, we found that GAiA-Net achieved the best performance when k was set to 4 on synthetic and real noise image datasets (CBSD68 and SIDD). In addition, we also observed that increasing k below or beyond a suitable number of k-nearest neighbors led to a decrease in performance, which may be because insufficient k-nearest neighbors result in unsatisfactory node representation, or redundant k-nearest neighbors cause node representation prone to over-fitting issues. Therefore, we set the number of k-nearest neighbors $k = 4$ in GAiA-Net for building graph features from feature maps.

Effect of Our Graph Block. To fairly compare the performances of all the methods, we set the parameter numbers of Case.II, Case.III, and the proposed GAiA-Net to be consistent. As shown in Fig. 11, compared with Case.II, the proposed GAiA blocks in our GAiA-Net not only reduces FLOPs and Runtime by **4.30**% and **2.46**%, but also improves PSNR by **0.18 dB** and **1.04 dB** on CBSD68 and SIDD validation datasets, respectively. This shows that the proposed GAiA block can significantly improve both the efficiency and the denoising performance.

To more intuitively illustrate the long-range dependencies on both the pixel-level and structure-level features retrieved by the GAiA block, we visualize the noise feature maps of Case.II, Case.III, and the GAiA block, respectively, which is shown in Fig. 12. Compared with the noise feature maps of Case.II and Case.III, the noise feature maps generated by GAiA block have obvious dense noise characteristics, which shows that the attention mechanism in GAiA block can capture more comprehensive noise from the perspective of pixel-level and structure-level features, thereby improving the quality of denoised images. Furthermore, the proposed GAiA-Net can significantly preserve and reconstruct finer details in denoised images. This suggests that long-range dependencies captured from pixel-level and structure-level features can facilitate a positive impact on reconstructing denoised images.

Effect of Loss Function. To investigate the impact of loss functions on the image denoising performance of GAiA-Net, we visualized the MAE (L1) loss function, the MSE (L2) loss function, and the Charbonnier loss function to observe their similarities and differences. As shown in Fig. 13, the Charbonnier loss function is more robust to outliers compared to the L1 loss function, which is highly sensitive to them.

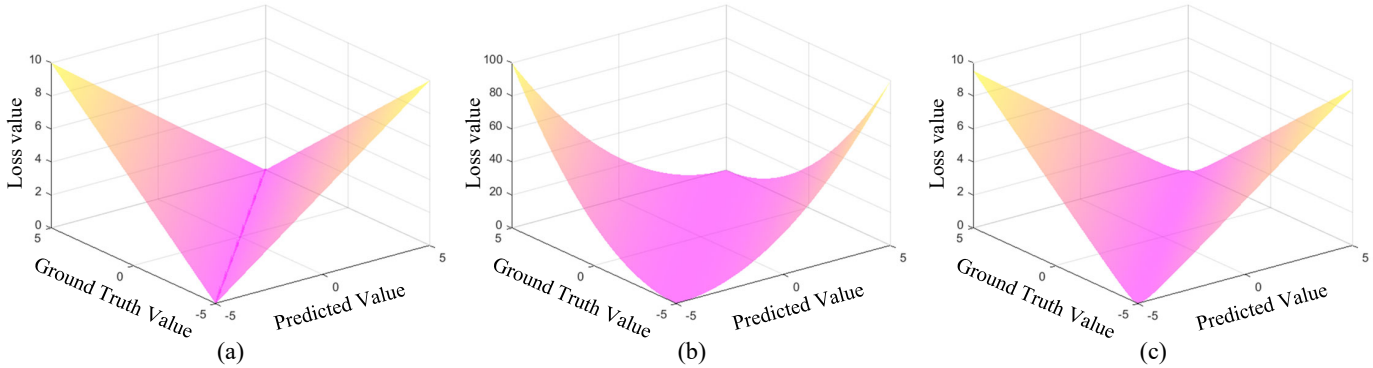

Fig. 13. Visualization of different loss functions. (a) Visualization of the MAE (L1) loss function. (b) Visualization of the MSE(L2) loss function. (c) Visualization of the Charbonnier loss function. In all three cases, the x-axis represents the difference between the predicted and ground truth values, while the y-axis represents the loss value. The steeper the slope of the curve, the more sensitive the loss function is to differences between predicted and ground truth values.

Additionally, the Charbonnier loss function is continuous and differentiable everywhere, making it easier to optimize using gradient-based methods, whereas the L2 loss function can produce unstable gradients when the predicted values are far from the ground truth. Therefore, the Charbonnier loss function strikes a balance between robustness and sensitivity, making it a great choice for image reconstruction tasks. To further validate the advantage of the Charbonnier loss function in the image denoising performance of GAiA-Net, we applied different loss functions to GAiA-Net and evaluated their image denoising performance, as shown in Table VII. Compared to MAE (L1) loss function and the MSE (L2) loss function, the Charbonnier loss function achieved the highest gain in image denoising performance. This indicates that the Charbonnier loss function is a more reliable and efficient choice that requires robustness and stability in the face of noisy images.

Discussion of Limitations. In this paper, we primarily investigate the proposed GAiA-Net for image denoising using graph representation, and evaluate it on both synthetic and real noise datasets. Future research should explore more tasks and further extend the method to video denoising. Additionally, although GAiA-Net effectively converts image feature maps to graph features for image denoising, our approach still has limitations in terms of flexibility compared to directly converting images to graph data. Therefore, future research should focus on addressing this issue.

V. Conclusion

In this paper, we propose **G**raph **A**ttention **i**n **A**ttention **Net**work (GAiA-Net) for the image denoising task. GAiA-Net first applies the novel graph construction to establish the graph, which can comprehensively and hierarchically transforming different information within (pixel-level) and outside (structure-level) of the nodes. In addition, to reconstruct the denoised image, long-range dependencies on both the pixel-level and structure-level features are captured using the proposed GAiA block. The proposed GAiA can first produce pixel-level attention within nodes, and then induce them to the retrieved structure-level feature to generate the final attention. Such long dependencies on both pixel and structure levels can significantly remove the complex noise from multiple noise sources. From extensive and comprehensive experiments on multiple denoising tasks, the proposed GAiA-Net achieves the state-of-the-art results quantitatively and qualitatively, demonstrating the satisfactory superiority on image denoising. We hope this pioneering GAiA-Net structure can encourage further exploration of architecture retrieving both pixel-level and structure-level features for image denoising tasks.

References

[1] G. Chen, H. Wang, K. Chen, Z. Li, Z. Song, Y. Liu, W. Chen, and A. Knoll, "A survey of the four pillars for small object detection: Multiscale representation, contextual information, super-resolution, and region proposal," *IEEE Transactions on systems, man, and cybernetics: systems*, 2020.

[2] W. Yang, T. Yuan, W. Wang, F. Zhou, and Q. Liao, "Single-image super-resolution by subdictionary coding and kernel regression," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 9, pp. 2478–2488, 2016.

[3] Y. Zhang, Y. Zhang, J. Zhang, D. Xu, Y. Fu, Y. Wang, X. Ji, and Q. Dai, "Collaborative representation cascade for single-image super-resolution," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 5, pp. 845–860, 2017.

[4] J. Mairal, F. R. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," *2009 IEEE 12th International Conference on Computer Vision*, pp. 2272–2279, 2009.

[5] A. Shocher, N. Cohen, and M. Irani, ""zero-shot" super-resolution using deep internal learning," in *CVPR*, 2018.

[6] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, pp. 3142–3155, 2017.

[7] K. Zhang, W. Zuo, and L. Zhang, "Ffdnet: Toward a fast and flexible solution for cnn-based image denoising," *IEEE Transactions on Image Processing*, vol. 27, pp. 4608–4622, 2018.

[8] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, "Swinir: Image restoration using swin transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1833–1844.

[9] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, and M.-H. Yang, "Restormer: Efficient transformer for high-resolution image restoration," *arXiv preprint arXiv:2111.09881*, 2021.

[10] D. Valsesia, G. Fracastoro, and E. Magli, "Deep graph-convolutional image denoising," *IEEE Transactions on Image Processing*, vol. 29, pp. 8226–8237, 2020.

[11] Y. Li, X. Fu, and Z. Zha, "Cross-patch graph convolutional network for image denoising," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4631–4640, 2021.

[12] Z. Wang, X. Cun, J. Bao, and J. Liu, "Uformer: A general u-shaped transformer for image restoration," *arXiv preprint arXiv:2106.03106*, 2021.

[13] C. Mou, J. Zhang, and Z. Wu, "Dynamic attentive graph learning for image restoration," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4308–4317, 2021.

[14] K. Zhang, Y. Li, J. Liang, J. Cao, Y. Zhang, H. Tang, R. Timofte, and L. Van Gool, "Practical blind denoising via swin-conv-unet and data synthesis," *arXiv preprint arXiv:2203.13278*, 2022.

[15] B. Jiang, Y. Lu, J. Wang, G. Lu, and D. Zhang, "Deep image denoising with adaptive priors," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, pp. 5124–5136, 2022.

[16] B. Jiang, Y. Lu, G. Lu, and D. Zhang, "Real noise image adjustment networks for saliency-aware stylistic color retouch," *Knowl. Based Syst.*, vol. 242, p. 108317, 2022.

[17] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, pp. 60–65 vol. 2, 2005.

[18] K. Dabov, A. Foi, V. Katkovnik, and K. O. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, pp. 2080–2095, 2007.

[19] S. Anwar and N. Barnes, "Real image denoising with feature attention," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3155–3164, 2019.

[20] B. Jiang, J. Wang, Y. Lu, G. Lu, and D. Zhang, "Multilevel noise contrastive network for few-shot image denoising," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–13, 2022.

[21] B. Jiang, Y. Lu, G. Lu, and D. Zhang, "Few-shot learning for image denoising," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2023.

[22] P. Liu, H. Zhang, K. Zhang, L. Lin, and W. Zuo, "Multi-level wavelet-cnn for image restoration," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 886–88 609, 2018.

[23] J.-J. Huang and P. L. Dragotti, "Winnet: Wavelet-inspired invertible network for image denoising," *IEEE Transactions on Image Processing*, vol. 31, pp. 4377–4392, 2022.

[24] A. Kolesnikov, A. Dosovitskiy, D. Weissenborn, G. Heigold, J. Uszkoreit, L. Beyer, M. Minderer, M. Dehghani, N. Houlsby, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021.

[25] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao, "Pre-trained image processing transformer," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12 294–12 305, 2021.

[26] K. Zhang, Y. Li, J. Liang, J. Cao, Y. Zhang, H. Tang, R. Timofte, and L. V. Gool, "Practical blind denoising via swin-conv-unet and data synthesis," *ArXiv*, vol. abs/2203.13278, 2022.

[27] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9992–10 002, 2021.

[28] B. Jiang, J. Li, H. Li, R. Li, D. Zhang, and G. Lu, "Enhanced frequency fusion network with dynamic hash attention for image denoising," *Information Fusion*, vol. 92, pp. 420–434, 2023.

[29] S. Zhou, J. Zhang, W. Zuo, and C. C. Loy, "Cross-scale internal graph neural network for image super-resolution," *ArXiv*, vol. abs/2006.16673, 2020.

[30] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[31] K. Han, Y. Wang, J. Guo, Y. Tang, and E. Wu, "Vision GNN: an image is worth graph of nodes," *CoRR*, vol. abs/2206.00272, 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2206.00272

[32] G. Li, M. Muller, A. Thabet, and B. Ghanem, "Deepgcns: Can gcns go as deep as cnns?" in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9267–9276.

[33] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 2. Ieee, 2005, pp. 60–65.

[34] B. Jiang, Y. Lu, J. Wang, G. Lu, and D. Zhang, "Deep image denoising with adaptive priors," *IEEE Transactions on Circuits and Systems for Video Technology*, 2022.

[35] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Transactions on Computational Imaging*, vol. 3, pp. 47–57, 2017.

[36] I. Loshchilov and F. Hutter, "Fixing weight decay regularization in adam," *ArXiv*, vol. abs/1711.05101, 2017.

[37] S. Roth and M. J. Black, "Fields of experts: a framework for learning image priors," *2005 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 860–867 vol. 2, 2005.

[38] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5197–5206, 2015.

[39] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep cnn denoiser prior for image restoration," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2808–2817, 2017.

[40] D. Liu, B. Wen, Y. Fan, C. C. Loy, and T. S. Huang, "Non-local recurrent network for image restoration," *ArXiv*, vol. abs/1806.02919, 2018.

[41] Y. Zhang, K. Li, K. Li, B. Zhong, and Y. R. Fu, "Residual non-local attention networks for image restoration," *ArXiv*, vol. abs/1903.10082, 2019.

[42] X. Jia, S. Liu, X. Feng, and L. Zhang, "Focnet: A fractional optimal control network for image denoising," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6047–6056, 2019.

[43] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. V. Gool, and R. Timofte, "Plug-and-play image restoration with deep denoiser prior," *IEEE transactions on pattern analysis and machine intelligence*, vol. PP, 2021.

[44] W. Li, X. Lu, J. Lu, X. Zhang, and J. Jia, "On efficient transformer and image pre-training for low-level vision," *ArXiv*, vol. abs/2112.10175, 2021.

[45] R. Franzen, "Kodak lossless true color image suite," *source: http://r0k.us/graphics/kodak*, vol. 4, no. 2, 1999.

[46] S. M. Kasar and S. Ruikar, "Image demosaicking by nonlocal adaptive thresholding," *2013 International Conference on Signal Processing , Image Processing & Pattern Recognition*, pp. 34–38, 2013.

[47] C. Tian, Y. Xu, and W. Zuo, "Image denoising using deep cnn with batch renormalization," *Neural networks : the official journal of the International Neural Network Society*, vol. 121, pp. 461–473, 2020.

[48] A. Abdelhamed, S. Lin, and M. S. Brown, "A high-quality denoising dataset for smartphone cameras," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1692–1700, 2018.

[49] J. Xu, H. Li, Z. Liang, D. C. Zhang, and L. Zhang, "Real-world noisy image denoising: A new benchmark," *ArXiv*, vol. abs/1804.02603, 2018.

[50] S. Nam, Y. Hwang, Y. Matsushita, and S. J. Kim, "A holistic approach to cross-channel image noise modeling and its application to image denoising," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1683–1691, 2016.

[51] J. Xu, L. Zhang, and D. D. Zhang, "A trilateral weighted sparse coding scheme for real-world image denoising," *ArXiv*, vol. abs/1807.04364, 2018.

[52] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang, "Toward convolutional blind denoising of real photographs," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1712–1722, 2019.

[53] Z. Yue, H. Yong, Q. Zhao, L. Zhang, and D. Meng, "Variational denoising network: Toward blind noise modeling and removal," in *NeurIPS*, 2019.

[54] R. Ma, B. Zhang, Y. Zhou, Z. Li, and F. Lei, "Pid controller-guided attention neural network learning for fast and effective real photographs denoising." *IEEE transactions on neural networks and learning systems*, vol. PP, 2021.

[55] Y. Kim, J. W. Soh, G. Y. Park, and N. I. Cho, "Transfer learning from synthetic to real-noise denoising with adaptive instance normalization," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3479–3489, 2020.

[56] S. W. Zamir, A. Arora, S. H. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao, "Learning enriched features for real image restoration and enhancement," *ArXiv*, vol. abs/2003.06792, 2020.

[57] R. Ma, S. Li, B. Zhang, and Z. Li, "Towards fast and robust real image denoising with attentive neural network and pid controller," *IEEE Transactions on Multimedia*, pp. 2366–2377, 2022.

[58] Y. Zhang, Y. Zhu, E. Nichols, Q. Wang, S. Zhang, C. Smith, and S. Howard, "A poisson-gaussian denoising dataset with real fluorescence microscopy images," *arXiv preprint arXiv:1812.10366*, 2018.